

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-227402

(43)Date of publication of application : 24.08.2001

(51)Int.Cl.

F02D 45/00

G06F 11/10

G06F 12/16

(21)Application number : 2000-035229

(71)Applicant : DENSO CORP

(22)Date of filing : 14.02.2000

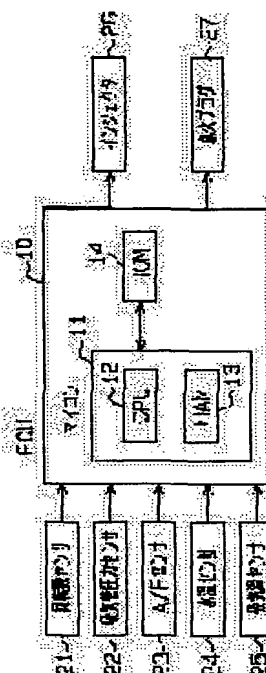
(72)Inventor : KONDO HIROSHI

(54) ON-VEHICLE ELECTRONIC CONTROL DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To efficiently operate a microcomputer and shorten a time till completion of calculation of a check sum.

SOLUTION: An ECU 10 is provided with a microcomputer 11 consisting of a CPU 12, RAM 13, etc., and a ROM 14 storing a control program, a data for deciding comparison, etc. The CPU 12 executes various engine controls such as fuel injection control and ignition timing control. The CPU 12 calculates a check sum for a specified address of the ROM 14. This check sum calculation is performed by time division, a number of addition bites is changed in accordance with a processing load of the CPU 12. For instance, a number of the addition bites is increased at low rotation time of an engine, in reverse, decreased at high rotation time of the engine.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

特開2001-227402

(2)

1

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-227402
(P2001-227402A)

(43) 公開日 平成13年8月24日(2001.8.24)

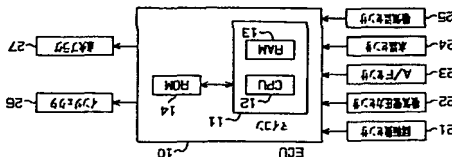
(51) Int. Cl. ⁷	F 02 D 45/00	F 1	チロート(参考)
F 02 D 45/00	3 72	F 02 D 45/00	3 72 F 3 C 0 8 4
			3 72 C 5 B 0 0 1
G 0 6 F 11/10	3 30	G 0 6 F 11/10	3 30 K 5 B 0 1 8
12/16	3 20	12/16	3 20 B 9 A 0 0 1

審査請求 米澤 光澤 請求項の数 6 OL (全 9 頁)

(21) 出願番号	特開2000-35226(P2000-35226)	(71) 出願人	00000490 株式会社デンソー
(22) 出願日	平成12年2月14日(2000.2.14)	(72) 発明者	近藤 裕 愛知県刈谷市昭和町1丁目1番地 株式会社デンソー内
		(74) 代理人	10068755 弁理士 豊田 博宣 (J41名) Fターム(参考) 30B4 B413 B415 B417 C409 D406 E502 E503 E506 F402 F411 F420 F425 F428 F433 5B01 A14 D203 5B018 G01 B413 J429 N404 R411 8A01 R806 E302 H34

(54) 発明の名称 車載電子制御装置

(57) 【要約】
【課題】マイクロコンピュータを効率的に動作させ、且つチェックサムの算出完了までの時間を短縮する。
【解決手段】ECU 10は、CPU 12、RAM 13等からなるマイコン11と、制御プログラムと比較判定用データ等を格納したROM 14とを備える。CPU 12は、燃料噴射制御や点火時期制御といった各種エンジン制御を実行する。また、CPU 12は、ROM 14の格納されたアドレスについてチェックサムを算出する。このチェックサム算出は時分で行い、CPU 12の処理負荷に応じて加算バイト数を大きくし、逆に、エンジンの高回転時には加算バイト数を小さくする。



(1) 特許請求の範囲

【請求項1】エンジン又は車載機器の各種制御に関する処理を定期的に実行するマイクロコンピュータを接続し、該定期的に実行される処理はマイクロコンピュータがメモリ内のチェックサムを分割して算出する車載電子制御装置であり、
マイクロコンピュータの処理負荷に応じて、チェックサムの算出時に1度に加算するバイト数を変更することを特徴とする車載電子制御装置。

【請求項2】エンジン回転数をモニタする手段を設け、該回転数が高いほど、チェックサム算出時の加算バイト数を小さくする請求項1に記載の車載電子制御装置。

【請求項3】エンジン回転数においてマイクロコンピュータがペーサ処理を繰り返している間、該ペーサ処理の1回当たりの所要時間を計測し、その所要時間が大きいほど、チェックサム算出時の加算バイト数を小さくする請求項1に記載の車載電子制御装置。

【請求項4】エンジン回転数又は車両走行状態をモニタし、その状態がマイクロコンピュータの処理負荷を減じる状態にあれば、チェックサム算出時の加算バイト数を大きくする請求項1に記載の車載電子制御装置。
【請求項5】各種制御に関する複数の処理について、実行されるエンジン回転数領域が各々相違する車載電子制御装置であって、
処理されるエンジン回転数領域が切り換えられるエンジン回転数に
対応して、チェックサムの算出時に1度に加算するバイト数を変更する請求項1に記載の車載電子制御装置。

【請求項6】エンジンの所定の高回転域では、チェックサムの算出を禁止する請求項1に記載の車載電子制御装置。
【発明の詳細な説明】
【0001】
【発明の属する技術分野】本発明は、エンジン又は車載機器の各種制御を実行する微成と、メモリ内のデータを加算してチェックサム(メモリデータ加算値)を算出する機構とを併せ持つ車載電子制御装置に関する。

【0002】
【従来の技術】従来より、メモリの異常検出や不正改変防止(タンバリング防止)を目的として、メモリのチェックサムを算出する車載電子制御装置が各種提案され、具体化されている。例えば、工場出荷時のメモリ検査等でメモリのチェックサムを算出し、そのチェックサムが真のサム値であるかどうかを判定してメモリの異常検出を行う技術が知られている。また、一般市場において、車両情報の一部としてメモリのチェックサムを算出し、そのチェックサムからメモリの不正改変を検出する技術が知られている。なお今後は、メモリのチェックサム値を所定のダイアグナスタに出力することが法規化される可能性がある(ISO15031-5のmode S0

9)。

【0003】ところで、車載電子制御装置(車載ECU)にはマイクロコンピュータが搭載され、該マイクロコンピュータ内のCPUにより各種のエンジン制御が実行され、且つ、チェックサム算出が実行される。この場合、エンジン制御に加えてチェックサム算出が実行されると、CPU処理負荷の増加をもたらす。それ従って、イグニッションキー(IGキー)のON直後におけるエンジン始動時、
・同IGキーのOFF直後におけるメインリレー制御時、
等々、エンジン制御によるCPUの処理負荷が比較的小さい状態でチェックサムが算出される。

【0004】上記の通りエンジン始動時にチェックサムを算出する場合、メモリ容量が比較的小さくチェックサム算出の所要時間が短ければチェックサムの算出完了までエンジン制御の開始を遅らせてもよい。しかしながら、各種電子制御の高機能化に伴いメモリ容量が増大する傾向にあることを考えると、チェックサム算出の所要時間が増加し、その分、エンジン制御の開始時間も遅延される。従って、エンジン始動性が悪化することが懸念される。

【0005】また、メインリレー制御中にチェックサムを算出する場合、IGキーがOFFされるまでの期間にチェックサムが算出され、メモリ異常があってもIGキーOFFまではそれが検出できないという不都合がある。
【0006】そこで近年では、上記の如くエンジン始動時やメインリレー制御時にチェックサムを算出する技術に代えて、通常のエンジン運転途中において、時分けてチェックサムを算出する技術が提案されている。つまり、時分けてチェックサムを算出してCPUの処理負荷を分散させることにより、エンジン制御に並行してチェックサム算出を実施できるようにしている。

【0007】時分けてチェックサムを算出する場合においても、エンジン回転数NEとCPU使用率との関係を図10に示す。なお、図10中、全CPU使用率のうち、エンジン制御によるCPU使用率を「A」で表し、チェックサム算出によるCPU使用率を「B」で表す。CPU使用率とは、処理されるべきタスクが単位時間内に占める比率であり、タスクの全処理時間と単位時間とが一致する時、CPU使用率=100%であると定義される。

【0008】図10(a)によれば、エンジン制御によるCPU使用率(図のA)は、エンジン回転数NEの増加に伴いほぼ比例的に増えるのに対し、チェックサム算出によるCPU使用率(図のB)は、エンジン回転数NEにほぼ一定となる。この場合、全CPU使用率(図のA+B)が100%に達する回転数が処理負荷の回転数であり、全CPU使用率が100%を超えると、本

記載であり、全CPU使用率が100%を超えると、本

記載であり、全CPU使用率が100%を超えると、本

記載であり、全CPU使用率が100%を超えると、本

記載であり、全CPU使用率が100%を超えると、本

記載であり、全CPU使用率が100%を超えると、本

【0030】図3の処理により、ROM14の加算開始アドレスから加算終了アドレスまでを一定分として、チェックサムが均分分割して算出され、その値に逐次、ROM全体のチェックサムが算出される。

【0031】図8には、時間割によるCPU使用率とエンジン回転数NEとの関係を示す。なお図8中、全CPU使用率のうち、エンジン制御によるCPU使用率を「A」で表し、チェックサム算出によるCPU使用率を「B」で表す。

【0032】図8によれば、エンジン制御によるCPU使用率（図8のA）は、エンジン回転数NEの増加に伴いはほぼ比例的に増える。つまり、エンジン高回転域ではほぼ比例的に増加し、CPU使用率が増加する。これに対し、チェックサム算出によるCPU使用率（図8のB）は、エンジン回転数NEの増加に伴い段階的に減じられている。前記図10（a）の従来技術と比較すると、図8では、エンジン制御によるCPU使用率が低い低回転域において、チェックサム算出によるCPU使用率が上がり、逆にエンジン制御によるCPU使用率が低い高回転域において、チェックサム算出によるCPU使用率が減らされることとなる。要するに、CPU使用率の増減が100%に対する余裕に比べて加算バイト数が設定されることとなり、回転数の全域でCPU使用率が向上するよう加算バイト数が最適化される。また図8では、前記図10（b）に示す従来技術とは異なり、システムにおける処理負担回転数が低下することとなる。

【0033】図7は、所定の進行パターンにおけるチェックサム算出時間を示すタイムチャートである。図7において、（a）は進行パターン上のエンジン回転数NEの変化をモデル化して示し、（b）はNEに応じて可変に設定される加算バイト数を示し、（c）は前記（b）の加算バイト数を積算した累積バイト数を示し、（d）は加算バイト数一定とする場合（従来技術）の累積バイト数を比較して示す。

【0034】（d）に示す従来技術の場合、エンジン回転数NEの発生に比例した一定量の加算バイト数が毎回設定され、時刻11でチェックサム算出が開始された後、時刻13でチェックサム算出が完了する。すなわち、チェックサム算出完了には「13-11」の時間を要する。

【0035】これに対し、本実施の形態の場合、（b）に示すようにエンジン回転数NEの変化に応じて1回の加算バイト数がその範囲に設定される。この場合、時刻11でチェックサム算出が開始された後、時刻12でチェックサム算出が完了する。すなわち、チェックサム算出完了には「12-11」の時間を要する。

【0036】つまり、従来技術では、エンジン高回転域でのCPU使用率の制限があるため、これを基準に加算バイト数が比較的小さく設定されていたが、本実施の形

態では、特に低回転域での加算バイト数が大きくなるため、1度により計算できる容量が増え、チェックサムの算出完了時間が短縮できる。

【0037】以上詳述した本実施の形態によれば、以下に示す効果が得られる。CPU12（マイコン11）の処理負担に応じて加算バイト数を変更するので、単位時間当たりのCPU処理能力を考慮しつつチェックサム算出が実施され、CPU12の実際の使用率が上がり、その結果、CPU12を効率よく動作させ、且つチェックサムの算出完了までの時間を短縮することができる。

【0038】この場合、ROM14が大容量化した場合でも、比較的短時間でチェックサム算出が完了できる。また、CPUの処理能力が向上が強いられることがないため、コストアップを招くこともない。

【0039】（第2の実施の形態）次に、本実施の形態第2の実施の形態を説明する。但し、本実施の形態では、上述した第1の実施の形態と同等であるものは説明を省略し、第1の実施の形態との相違点を中心に説明する。

【0040】上記第1の実施の形態では、エンジン制御によるCPU処理負担（使用率）がエンジン回転数NEに応じてほぼ比例的に上昇する旨を記載したが、実際のエンジン制御では、各種制御に関する複数の処理について、実施されるエンジン回転数領域を各々相違し、回転数領域に応じてCPU処理負担に変動が存する。つまり、一例として、低・中回転域（例えば0〜4000rpm）ではA/Fフィードバック制御が実施されるのに対し、高回転域（4000rpm以上）ではA/Fフィードバック制御が停止される。またその他、失火検出やアイドル回転制御の処理も低・中回転域でのみ実施される。

【0041】それ故、図8（a）に示すように、エンジン制御によるCPU使用率（図8のA）がエンジン回転数NEに応じて一様に上昇するのではなく、変動点を持つこととなる。

【0042】そこで本実施の形態では、上記図8（a）の図8に類似した加算バイト数を算出する。要するに、加算バイト数の算出に際し、前記図5の図5の代りに図9の図8（b）によれば、チェックサム算出によるCPU使用率（図8のB）は、エンジン回転数NE並びに制御の変動点に応じて段階的に変化する。この場合にもやはり、回転数の全域でCPU使用率が向上するよう加算バイト数が最適化される。

【0043】こうして図9の図8に類似した加算バイト数を決定し、チェックサム算出を行う場合、チェックサム算出も各々が全CPU使用率は図8（b）のようになる。図8（b）によれば、チェックサム算出によるCPU使用率（図8のB）は、エンジン回転数NE並びに制御の変動点に応じて段階的に変化する。この場合にもやはり、回転数の全域でCPU使用率が向上するよう加算バイト数が最適化される。

【0044】以上第2の実施の形態によれば、処理の実

施/非実施が切り換えられるエンジン回転数（回転数に定まる。この場合、エンジン高回転域において、チェックサム算出以外に本来実施される処理（タスク）が実施されない、或いは実施が遅れるといった不都合が解消される。【図面の簡単な説明】

【図1】発明の実施の形態における車載ECUの概要を示す構成図。

【図2】CPUの制御プログラムの概要を示すフローチャート。

【図3】チェックサムの算出手順を示すフローチャート。

【図4】加算バイト数の決定手順を示すフローチャート。

【図5】エンジン回転数と加算バイト数との関係を示す図。

【図6】エンジン回転数とCPU使用率との関係を示す図。

【図7】進行パターンにおけるエンジン回転数と加算バイト数との関係を示す図。

【図8】第2の実施の形態におけるエンジン回転数と加算バイト数との関係を示す図。

【図9】第2の実施の形態におけるエンジン回転数と加算バイト数との関係を示す図。

【図10】従来のエンジン回転数と加算バイト数との関係を示す図。

【図11】1度に加算するバイト数とチェックサムの算出完了時間との関係を示す図。

【図12】従来のエンジン回転数と加算バイト数との関係を示す図。

【図13】従来のエンジン回転数と加算バイト数との関係を示す図。

【図14】従来のエンジン回転数と加算バイト数との関係を示す図。

【図15】従来のエンジン回転数と加算バイト数との関係を示す図。

【図16】従来のエンジン回転数と加算バイト数との関係を示す図。

【図17】従来のエンジン回転数と加算バイト数との関係を示す図。

【図18】従来のエンジン回転数と加算バイト数との関係を示す図。

【図19】従来のエンジン回転数と加算バイト数との関係を示す図。

【図20】従来のエンジン回転数と加算バイト数との関係を示す図。

【図21】従来のエンジン回転数と加算バイト数との関係を示す図。

【図22】従来のエンジン回転数と加算バイト数との関係を示す図。

【図23】従来のエンジン回転数と加算バイト数との関係を示す図。

【図24】従来のエンジン回転数と加算バイト数との関係を示す図。

【図25】従来のエンジン回転数と加算バイト数との関係を示す図。

【図26】従来のエンジン回転数と加算バイト数との関係を示す図。

【図27】従来のエンジン回転数と加算バイト数との関係を示す図。

【図28】従来のエンジン回転数と加算バイト数との関係を示す図。

【図29】従来のエンジン回転数と加算バイト数との関係を示す図。

【図30】従来のエンジン回転数と加算バイト数との関係を示す図。

【図31】従来のエンジン回転数と加算バイト数との関係を示す図。

【図32】従来のエンジン回転数と加算バイト数との関係を示す図。

【図33】従来のエンジン回転数と加算バイト数との関係を示す図。

【図34】従来のエンジン回転数と加算バイト数との関係を示す図。

【図35】従来のエンジン回転数と加算バイト数との関係を示す図。

【図36】従来のエンジン回転数と加算バイト数との関係を示す図。

【図37】従来のエンジン回転数と加算バイト数との関係を示す図。

【図38】従来のエンジン回転数と加算バイト数との関係を示す図。

【図39】従来のエンジン回転数と加算バイト数との関係を示す図。

【図40】従来のエンジン回転数と加算バイト数との関係を示す図。

【図41】従来のエンジン回転数と加算バイト数との関係を示す図。

【図42】従来のエンジン回転数と加算バイト数との関係を示す図。

【図43】従来のエンジン回転数と加算バイト数との関係を示す図。

【図44】従来のエンジン回転数と加算バイト数との関係を示す図。

【図45】従来のエンジン回転数と加算バイト数との関係を示す図。

【図46】従来のエンジン回転数と加算バイト数との関係を示す図。

【図47】従来のエンジン回転数と加算バイト数との関係を示す図。

【図48】従来のエンジン回転数と加算バイト数との関係を示す図。

【図49】従来のエンジン回転数と加算バイト数との関係を示す図。

【図50】従来のエンジン回転数と加算バイト数との関係を示す図。

【図51】従来のエンジン回転数と加算バイト数との関係を示す図。

【図52】従来のエンジン回転数と加算バイト数との関係を示す図。

【図53】従来のエンジン回転数と加算バイト数との関係を示す図。

【図54】従来のエンジン回転数と加算バイト数との関係を示す図。

【図55】従来のエンジン回転数と加算バイト数との関係を示す図。

【図56】従来のエンジン回転数と加算バイト数との関係を示す図。

【図57】従来のエンジン回転数と加算バイト数との関係を示す図。

【図58】従来のエンジン回転数と加算バイト数との関係を示す図。

【図59】従来のエンジン回転数と加算バイト数との関係を示す図。

【図60】従来のエンジン回転数と加算バイト数との関係を示す図。

【図61】従来のエンジン回転数と加算バイト数との関係を示す図。

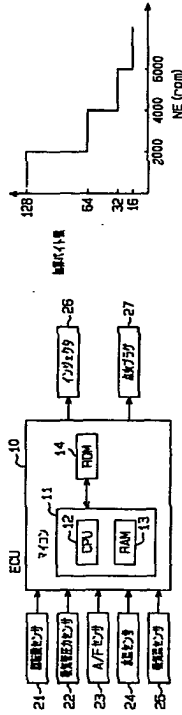
【図62】従来のエンジン回転数と加算バイト数との関係を示す図。

【図63】従来のエンジン回転数と加算バイト数との関係を示す図。

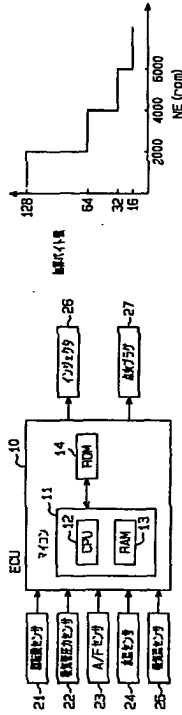
【図64】従来のエンジン回転数と加算バイト数との関係を示す図。

【図65】従来のエンジン回転数と加算バイト数との関係を示す図。

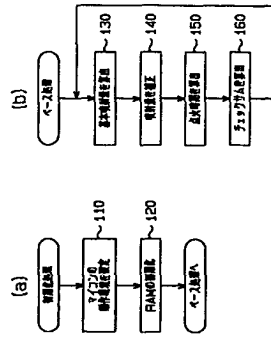
【図1】



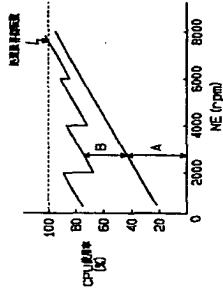
【図5】



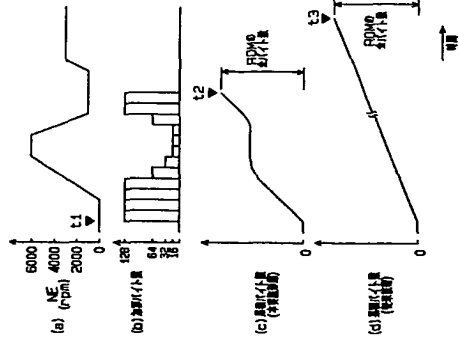
【図2】



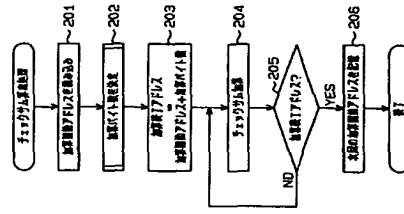
【図6】



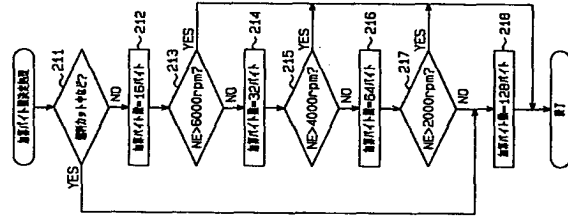
【図7】



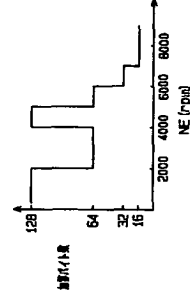
【図3】



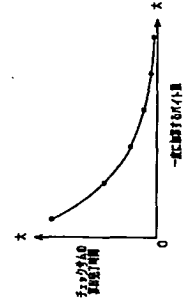
【図4】



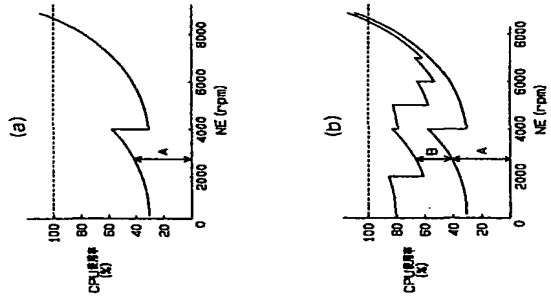
【図9】



【図11】



[図8]



[図10]

